

Introduzione a CSS - Parte 4

Ma quando finisce?

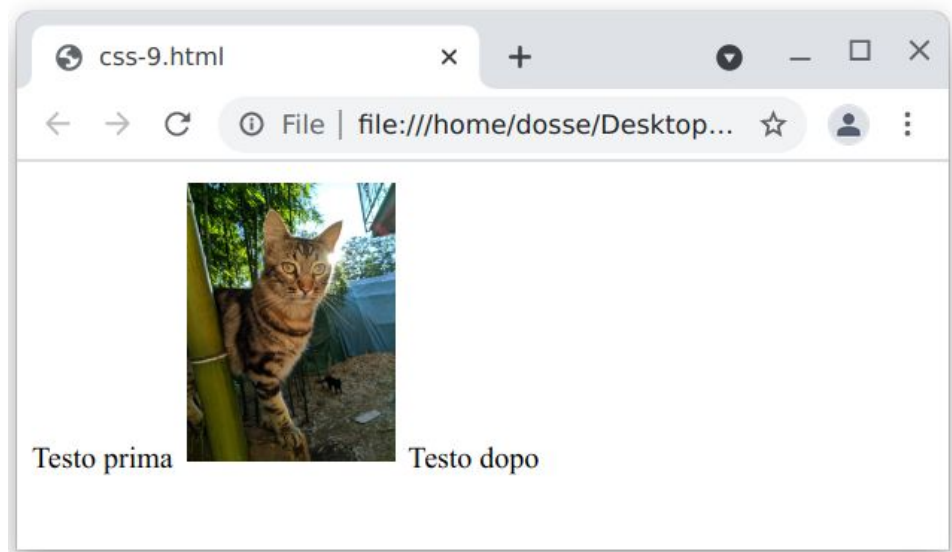
Parte 4

- Cosa vediamo?
 - La proprietà display
 - La proprietà position
 - Ombre
 - Bordi arrotondati
 - Media query

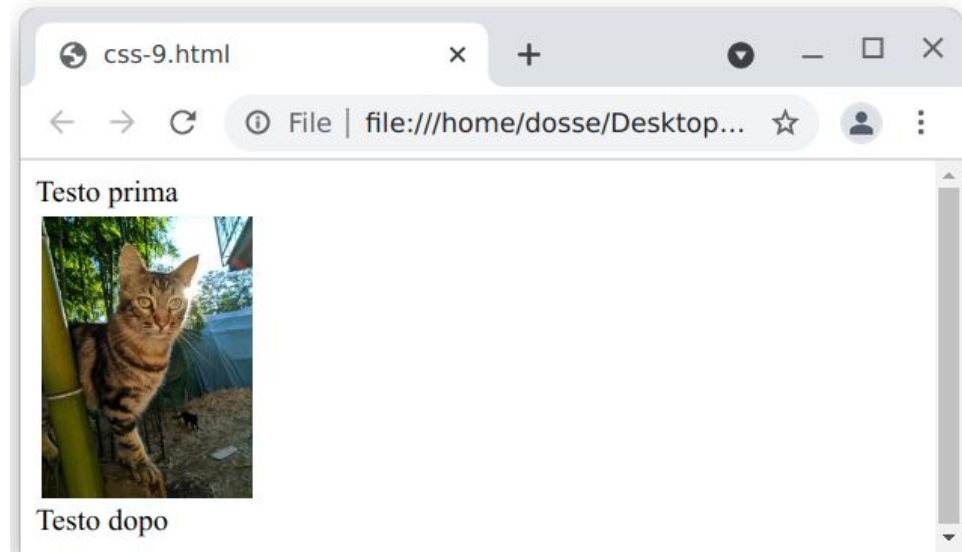
La proprietà display

- La proprietà display di CSS permette di cambiare il „comportamento“ di un elemento. Vediamo alcuni valori possibili:
 - **none**: è come se l'elemento non esistesse
 - **inline**: si comporta come se fosse del testo (si comporta così).
Proprietà tipo width e height non hanno effetto su elementi inline
 - **block**: occupa un'intera linea, andando a capo automaticamente prima e dopo (<div> e <p> si comportano così)
 - **inline-block**: si comporta come un inline, ma si possono usare width, height, ecc. (si comporta così). È come se fosse un „carattere gigante“

Esempio



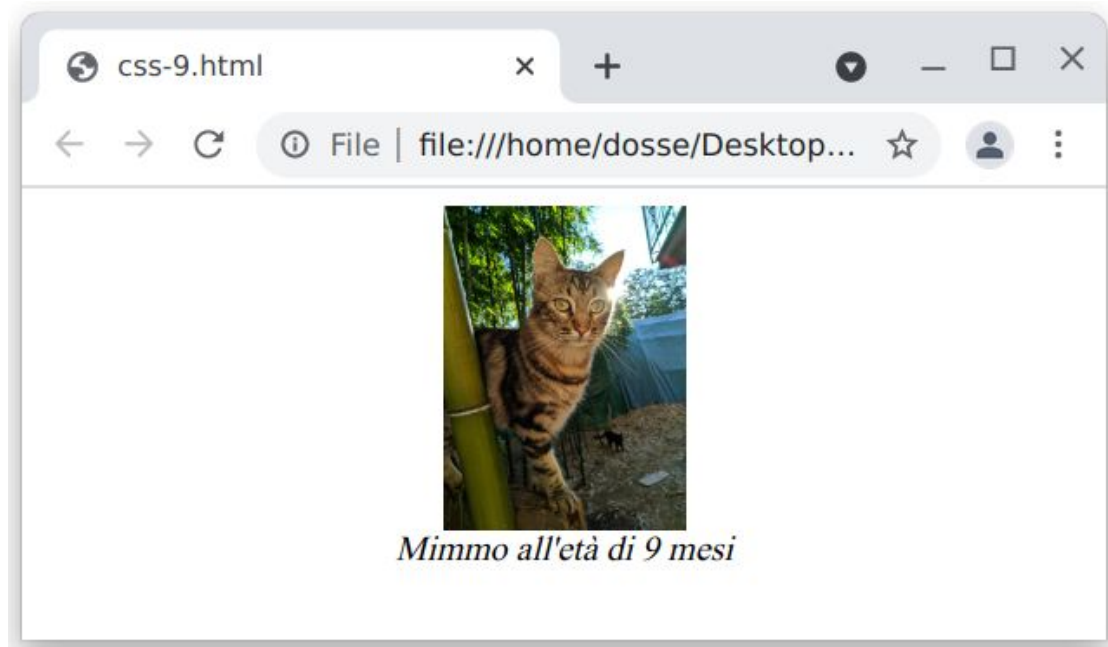
`` senza stile



`` con
`display: block;`

Esempio

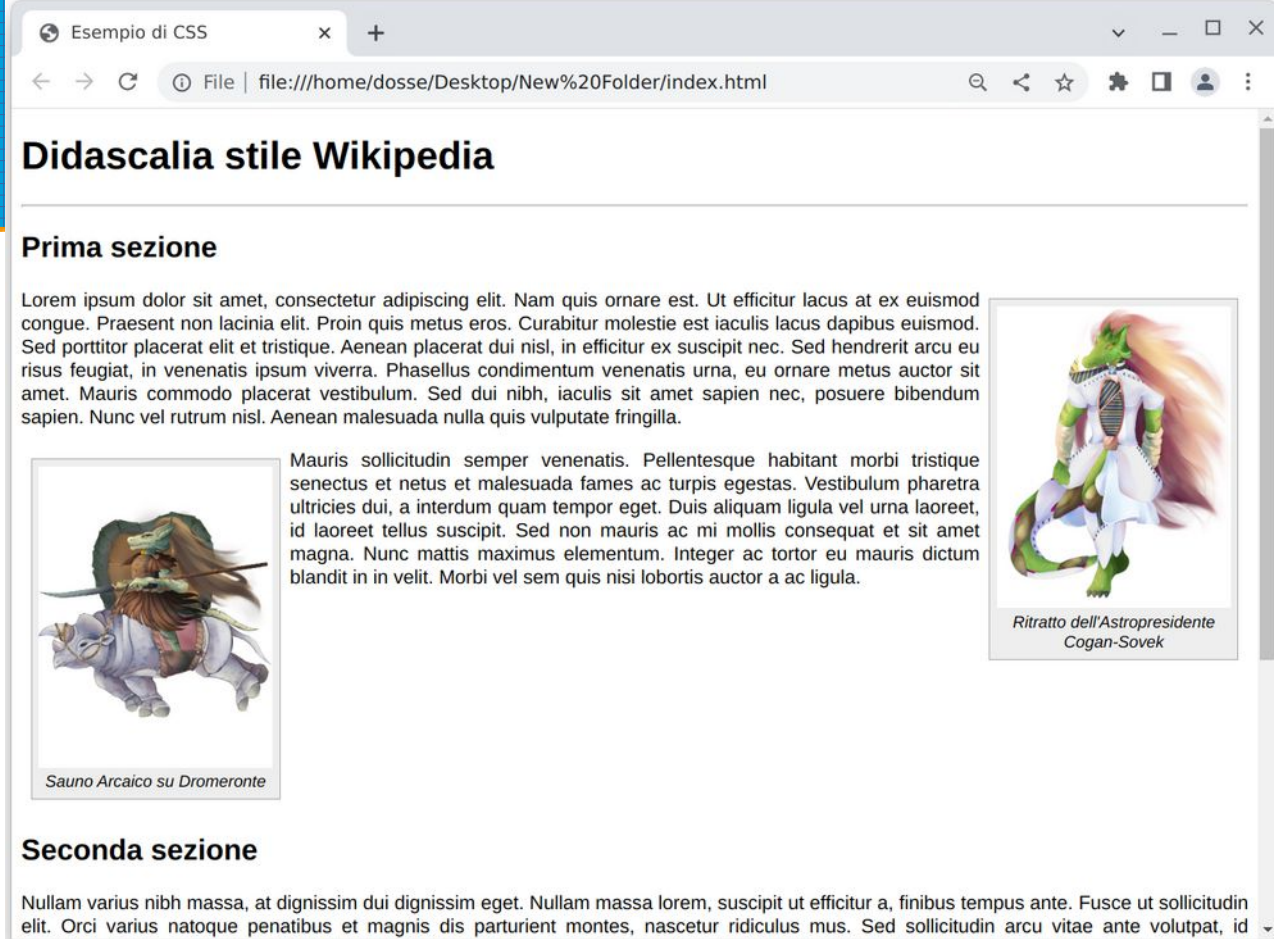
```
div.didascalia{  
    text-align:center;  
    font-style:italic;  
}  
div.didascalia img{  
    display:block;  
    margin:0 auto;  
}
```



```
<div class="didascalia">  
      
    Mimmo all'età di 9 mesi  
</div>
```

Esempio

```
div.didwiki{
  width:12rem;
  background:#EEEEEE;
  color:black;
  padding:0.35rem;
  margin:0.5rem;
  border:1px solid darkgray;
  font-size:0.85rem;
  font-style:italic;
  text-align:center;
}
div.didwiki > img{
  display:block;
  width:100%;
  height:auto;
  margin-bottom:0.2rem;
}
.fluttuaSX{
  float:left;
}
.fluttuaDX{
  float:right;
}
.barrieraFloat{
  clear:both;
}
```



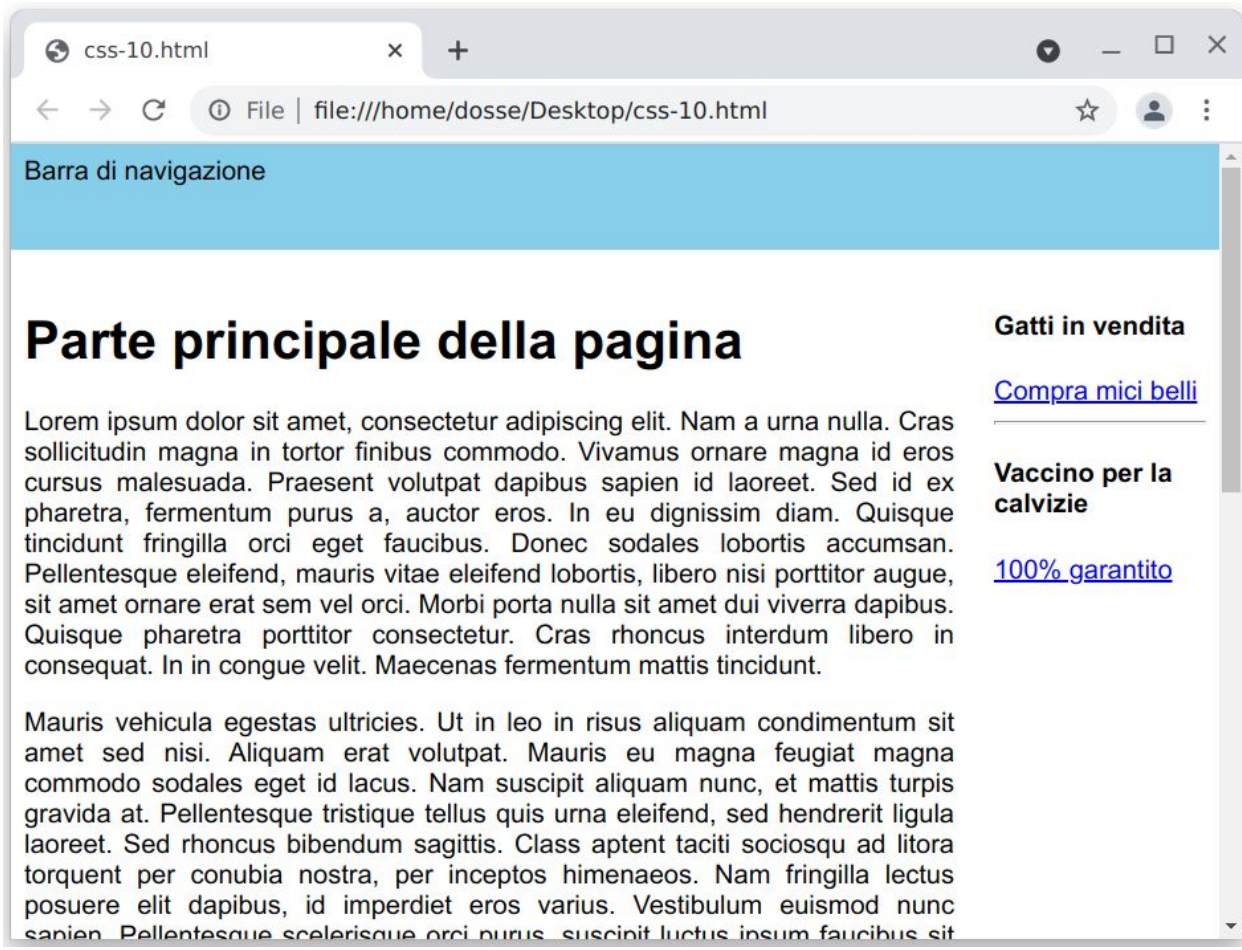
```
<div class="didwiki fluttuaDX">
  
  Ritratto dell'Astropresidente Cogan-Sovek
</div>
```

La proprietà position

- La proprietà position permette di specificare la modalità di posizionamento di un elemento
 - **static**: è il posizionamento di default, il browser decide dove mettere l'elemento
 - **relative**: permette di spostare l'elemento dalla sua posizione di default utilizzando attributi come top, bottom, left, right
 - **absolute**: permette di specificare manualmente la posizione dell'elemento utilizzando attributi come top, bottom, left, right
 - **fixed**: l'elemento è fisso sullo schermo, anche se la pagina scorre (utile per i menu)
 - altri...

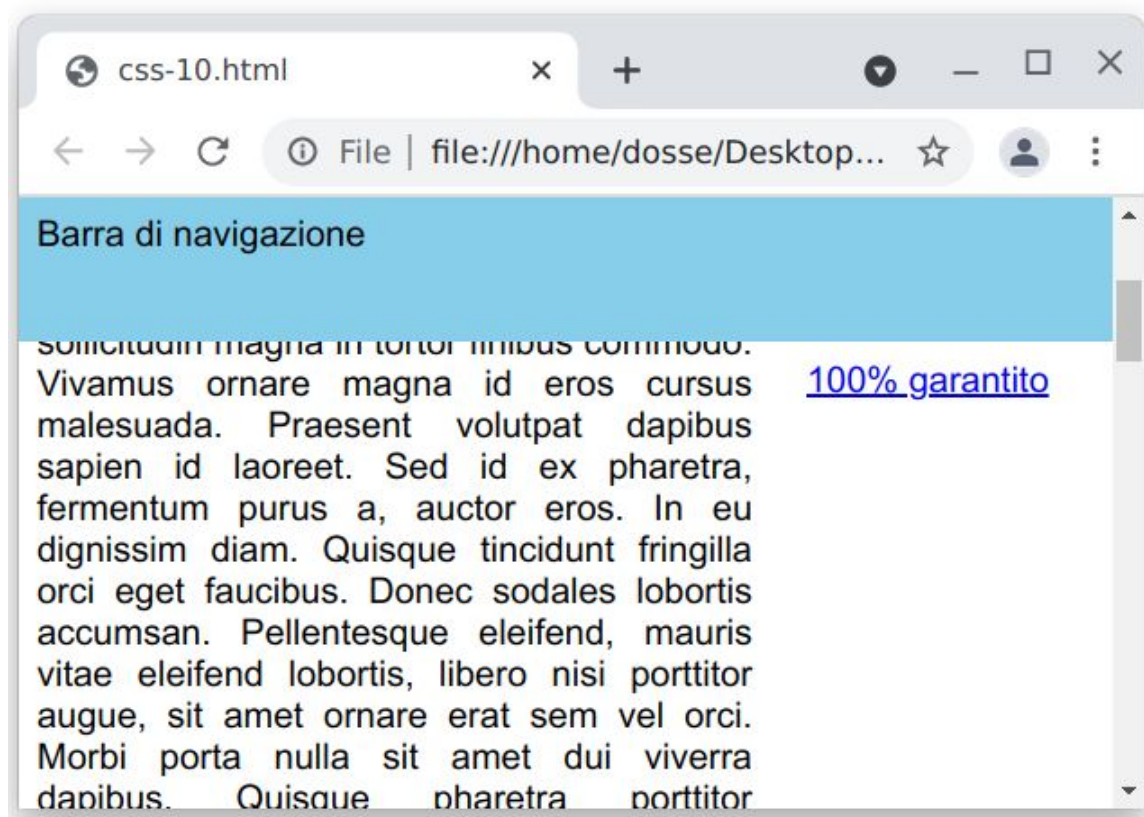
Esempio

```
body{
  margin:0;
  position:relative;
  font-family:sans-serif;
}
#navigazione{
  position:fixed;
  top:0; left:0;
  width:100%; height:3em;
  background:skyblue;
  padding:0.5em;
  z-index:99999;
}
#principale{
  position:absolute;
  top:5em; left:0.5em; right:10em;
  text-align:justify;
}
#ads{
  position:absolute;
  top:5em; right:0.5em;
  width:8em;
}
```



Esempio

Scorrendo la pagina, la barra di navigazione resta fissata in cima e davanti al contenuto



z-index?

```
#navigazione{  
  position:fixed;  
  top:0; left:0;  
  width:100%; height:3em;  
  background:skyblue;  
  padding:0.5em;  
  z-index:99999;  
}
```

Questa dichiarazione dice al browser che l'elemento deve essere disegnato davanti a tutto

Giocando con il valore di z-index su più elementi è possibile specificare l'ordine di disegno degli elementi che si sovrappongono

Ombre

- La proprietà box-shadow permette di aggiungere un'ombreggiatura (o un alone colorato) sotto a un elemento

```
box-shadow: 0em 0em 1em black;
```

Scostamento
orizzontale

Scostamento
verticale

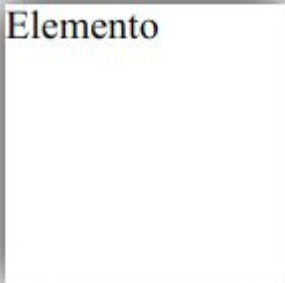
Raggio

Colore

Esempio

```
#esempio{  
  width:7em;  
  height:7em;  
  margin:5em;  
  box-shadow:0 0 1em black;  
}
```

Elemento



```
#esempio{  
  width:7em;  
  height:7em;  
  margin:5em;  
  box-shadow:0em 0em 0.5em black inset;  
}
```

Elemento



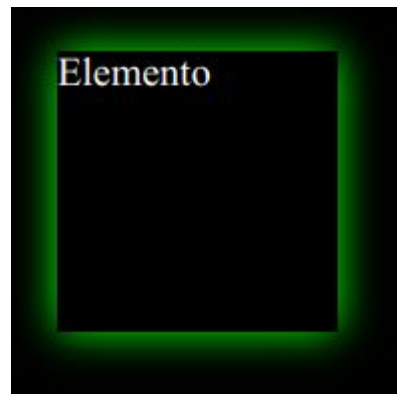
```
#esempio{  
  width:7em;  
  height:7em;  
  margin:5em;  
  box-shadow:0.2em 0.2em 1em black;  
}
```

Elemento



Esempio

```
body{  
    background:black;  
    color:white;  
}  
#esempio{  
    width:7em;  
    height:7em;  
    margin:5em;  
    box-shadow:0em 0em 1em lime;  
}
```



```
body{  
    background:black;  
    color:white;  
}  
#esempio{  
    width:7em;  
    height:7em;  
    margin:5em;  
    box-shadow:0em 0em 1em lime, 0em 0em 5em skyblue;  
}
```



Ombre

- La proprietà text-shadow funziona in modo analogo, ma per il testo

```
text-shadow: 0em 0em 0.2em black;
```

Scostamento
orizzontale

Scostamento
verticale

Raggio

Colore

Esempio

```
#esempio{  
  font-family:"Noto Sans";  
  font-size:4rem;  
  font-weight:lighter;  
  text-shadow:0 0 0.2em red;  
}
```

Elemento


```
#esempio{  
  font-family:"Noto Sans";  
  font-size:4rem;  
  font-weight:lighter;  
  text-shadow:0.05em 0.05em 0.1em black;  
}
```

Elemento

Bordi arrotondati

- La proprietà `border-radius` permette di arrotondare i bordi di un elemento

`border-radius: 1em;`



Raggio

Esempio

```
a.carino{
  display:inline-block;
  color:green;
  text-decoration:none;
  padding:0.3em;
  border:0.1em solid green;
  border-radius:0.3em;
}
```

Link bottonoso

```
body{
  background:black;
  color:white;
}
a.hackerino{
  display:inline-block;
  color:lime;
  font-family:monospace;
  text-decoration:none;
  padding:0.7em;
  border:0.1em solid lime;
  border-radius:9999em;
  box-shadow:0 0 0.7em lime;
  text-shadow:0 0 0.3em lime;
}
```

Hack system

Media query

- Le media query di CSS 3 consentono alla pagina di **adattarsi in base al dispositivo** che deve visualizzare la pagina
- L'idea di base è di **attivare/disattivare degli insiemi di regole in base a delle condizioni**
- Ad esempio si possono rimuovere gli sfondi se la pagina viene stampata, oppure si può visualizzare un layout diverso se la pagina è visualizzata da uno smartphone anziché da un PC

Media query

- Per segnalare al browser che la vostra pagina ha un CSS adattabile a dispositivi di dimensioni diverse, bisogna aggiungere questo tag nell'**head di ogni file HTML**:

```
<meta name="viewport" content="width=device-width" />
```

- Senza questo tag, il browser tenterà di adattare automaticamente la pagina al dispositivo e avrà un aspetto scorretto (di solito il testo è ingrandito e le immagini sono rimpicciolite)

Media query

- Per attivare un insieme di regole solo su carta, si può usare

```
@media print{  
    regole...  
}
```


Media query

- Per attivare un insieme di regole su display piccoli, si può usare

```
@media screen and (max-width: 60rem) {  
    regole...  
}
```

↑
Attenzione allo spazio!

- La regola si attiva solo se la finestra (o lo schermo) sono larghi meno di 60 caratteri

Media query

- Per attivare un insieme di regole su display grandi, si può usare

```
@media screen and (min-width: 60rem){  
    regole...  
}
```

- La regola si attiva solo se la finestra (o lo schermo) sono larghi almeno 60 caratteri

Media query

- Per attivare un insieme di regole su qualsiasi supporto che rispetta una certa condizione, si può usare

```
@media all and (max-width: 15cm){  
    regole...  
}
```

- La regola si attiva su qualsiasi supporto la cui larghezza è meno di 15cm

Media query

- Per attivare un insieme di regole se il dispositivo è in orizzontale

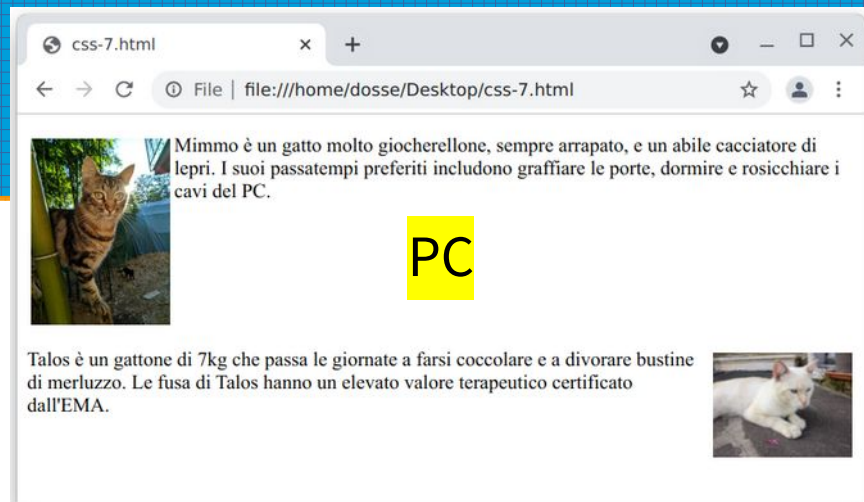
```
@media screen and (orientation: landscape){  
    regole...  
}
```

- La regola si attiva su qualsiasi dispositivo orientato in orizzontale (landscape=orizzontale, portrait=verticale)

Esempio

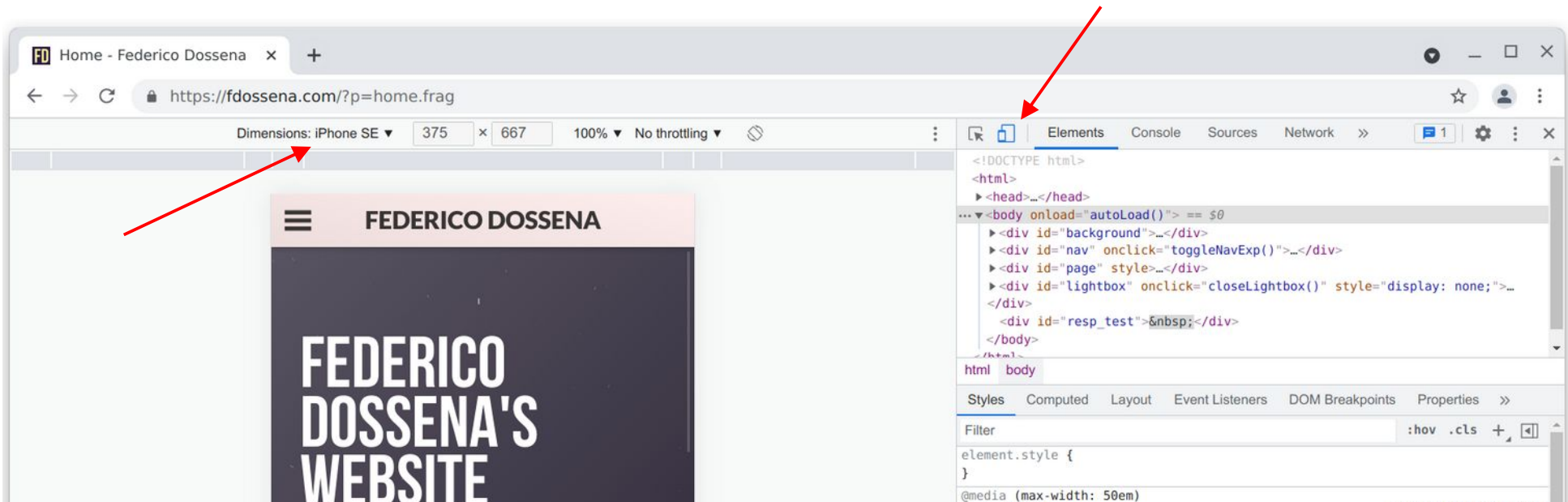
- Vogliamo che gli oggetti fluttuanti occupino l'intera riga su schermi piccoli

```
.fluttuaSX{
    float:left;
}
.fluttuaDX{
    float:right;
}
@media all and (max-width: 40rem){
    .fluttuaSX, .fluttuaDX{
        float:none;
        display:block;
        margin:0 auto;
    }
}
```



Media query

- Per testare le media query, negli strumenti di sviluppo del browser (tasto F12) è possibile attivare una modalità che simula il dispositivo mobile premendo questo pulsante e scegliendo un dispositivo da simulare



To be continued????

- Come introduzione quello che abbiamo visto può bastare, ma c'è molto che non abbiamo visto: animazioni, flexbox, interazioni, funzioni, ...
- Puoi trovare molto altro materiale, inclusi tutorial, esempi ed esercizi su W3Schools:
<https://www.w3schools.com/css/default.asp>